

PA01-308
reference 5

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-232875

(43) 公開日 平成10年(1998) 9月2日

(51) Int.Cl. ⁸	識別記号	F I	
G 0 6 F 17/30		G 0 6 F 15/40	3 1 0 B
12/00	5 1 3	12/00	5 1 3 D
	5 4 5		5 4 6 A
		15/403	3 3 0 B

審査請求 未請求 請求項の数 5 O L (全 15 頁)

(21) 出願番号 特願平9-35021
(22) 出願日 平成9年(1997) 2月19日

(71) 出願人 000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目6番地
(72) 発明者 小林 肇
神奈川県川崎市幸区鹿島田880番地の12
株式会社日立製作所情報・通信開発本部内
(72) 発明者 山本 祥一
神奈川県川崎市幸区鹿島田880番地の12
株式会社日立製作所情報・通信開発本部内
(72) 発明者 土田 正士
神奈川県川崎市幸区鹿島田880番地の12
株式会社日立製作所情報・通信開発本部内
(74) 代理人 弁理士 有近 紳吉郎

最終頁に続く

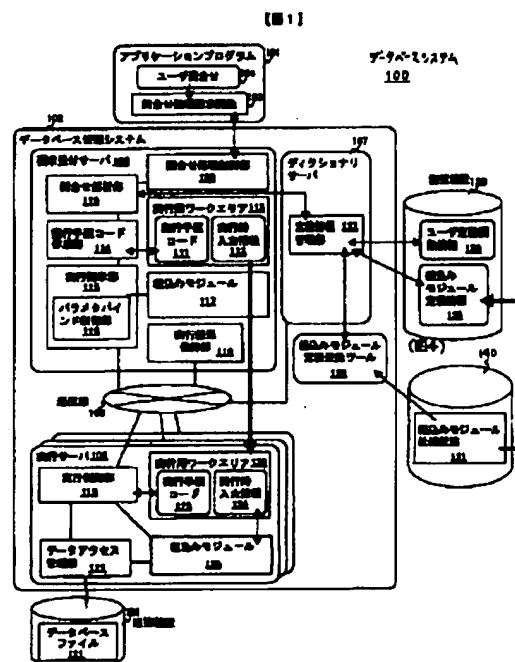
(54) 【発明の名称】 データベース管理方法および並列データベース管理システム

(57) 【要約】

【課題】 システム性能を向上させる。

【解決手段】 要求受付サーバ105上で組込みモジュール117を実行することによって実行時入力情報112を取得し、それを複数の実行サーバ106上の実行時入力情報124に複写し、その実行時入力情報124を利用して複数の実行サーバ106は組込みモジュール125を実行する。

【効果】 入力情報を取得する処理の重複や集中を防止でき、システム性能を向上できる。



【特許請求の範囲】

【請求項 1】 ユーザからの要求を受け付けてその要求に見合うデータベース処理の実行手順を指示する実行手順コードを作成する要求受付サーバと前記実行手順コードに従って前記データベース処理を並列に実行する複数の実行サーバとを有し、ユーザによって組み込まれる組み込みモジュールを実行する機能を前記要求受付サーバと前記実行サーバとが有する並列データベース管理システムにおいて、

前記実行サーバ上で組み込みモジュールを実行する際に必要な入力情報を前記要求受付サーバが取得すべきことを認識する解析過程と、前記認識に従って前記要求受付サーバ上で入力情報取得用の組み込みモジュールを実行して前記入力情報を取得する取得過程と、前記入力情報を渡す指示を含ませるように前記実行手順コードを編集する設定過程とを含むデータベース実行制御を行なうことを特徴とするデータベース管理方法。

【請求項 2】 請求項 1 に記載のデータベース管理方法において、組み込みモジュールのインタフェイス仕様の記述をもとに組み込みモジュールの実行制御を行なうことを特徴とするデータベース管理方法。

【請求項 3】 請求項 2 に記載のデータベース管理方法において、ユーザによる組み込みモジュールの機能呼び出しで渡される入力情報以外の付加的な入力情報を当該組み込みモジュールの実行のために渡す指示を当該組み込みモジュールのインタフェイス仕様の記述に含み得ることを特徴とするデータベース管理方法。

【請求項 4】 請求項 2 または請求項 3 に記載のデータベース管理方法において、前記実行サーバ上で組み込みモジュールを実行する際に必要な入力情報を前記要求受付サーバが入力情報取得用の組み込みモジュールを実行して取得すべきことを示す指示を前記実行サーバ上で実行する組み込みモジュールまたは前記入力情報取得用の組み込みモジュールの少なくとも一方のインタフェイス仕様の記述に含み得ることを特徴とするデータベース管理方法。

【請求項 5】 ユーザからの要求を受け付ける要求受付サーバとその要求受付サーバの指示に従ってユーザからの要求に見合うデータベース処理を並列に実行する複数の実行サーバとを有する並列データベース管理システムにおいて、

前記要求受付サーバが入力情報取得用の組み込みモジュールを実行して入力情報を取得し、前記実行サーバが前記入力情報を用いて組み込みモジュールを実行することとを特徴とする並列データベース管理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、システム性能を向上することが出来るデータベース管理方法および並列データベース管理システムに関する。

【0002】

【従来の技術】 図 10 に、従来のデータベースシステムの一例の構成概念図を示す。このデータベースシステム 1000 は、SGML (Standard Generalized Markup Language: 標準一般化マーク付け言語) 文書を管理するためのシステムであり、データベース管理システム 1102 と、記憶装置 1126 および 1128 とを具備している。

【0003】 前記データベース管理システム 1102 は、ユーザ問合せ 104 を受け付ける要求受付サーバ 1105 と、その要求受付サーバ 1105 からの指示に従ってデータベース処理を実行する複数の実行サーバ 1106 と、システムの定義情報を管理するディクショナリサーバ 1107 と、これらのサーバ 1105、1106、1107 を接続する通信路 108 とを具備している。

【0004】 SGML 文書の例として次の定義を用いる。

```
CREATE TYPE sgmltext_t (
    text BLOB,
    FUNCTION extract ( sgmltext_t,
                      VARCHAR )

    RETURNS BLOB
    LANGUAGE C
    EXTERNAL NAME 'p_sgml_extract';
);

CREATE TABLE reports (
    published_date DATE,
    contents sgmltext_t );
```

なお、上記の例は、SQL3 (Structured Query Language: 構造化問合せ言語) によって記述されている。

【0005】 ユーザは、SGML で記述されたデータ (以下、SGML テキストと略す) を ADT 型 (Abstract Data Type: 抽象データ型) の sgmltext_t 型を用いて管理する。この sgmltext_t 型は、BLOB (Binary Large Object) 型の text を構成要素とする。

【0006】 また、ユーザのデータベースモデルにおける「報告書」に対応するデータの構造をテーブル reports を用いて表わす。さらに、「報告書」が「発行日」と「報告内容」とを構成要素とすることに対応させて、テーブル reports が DATE 型のカラム published_date と sgmltext_t 型のカラム contents からなるように定義する。大量の SGML 文書を並列に処理できるように、テーブル reports のレコードおよび SGML テキストは、実行サーバ 1106 がアクセスする記憶装置 1126 に保持されている。なお、「報告書」を「発行日」による条件で高速検索できるように、実行サーバ 1106 が提供する一般的なインデクス機能を用いて、テーブル reports のカラム published_date にインデクスを設定する。

【0007】 ADT 関数 extract() は、SGML テキス

トの中からタグで区切られるテキストデータを抽出する関数であり、SGMLテキストおよび抽出する部分を指定するタグ名を入力情報とし、sgmltext_tの型を持つデータに対して適用でき、C言語で記述された外部関数p_sgml_extractに関連付けられている。

【0008】外部関数p_sgml_extract()は、前記ADT関数extract()を実現する関数として定義されている。前記実行サーバ1106には、SGML文書データ管理機能を実現するための組込みモジュール1125が組み込まれている。この組込みモジュール1125には、前記外部関数p_sgml_extract()を実現するためのコード1001が含まれている。

【0009】前記ディクショナリサーバ1107がアクセスする記憶装置128には、抽出パラメタ1002が保持されている。この抽出パラメタ1002は、指定されたタグで区切られるテキストデータをSGML文書から抽出するために用いられる制御情報である。また、抽出パラメタは、抽出したテキストデータをSGML文書として構造化するための構造情報を含んでおり、抽出結果を作成するためには不可欠な情報である。抽出パラメタは、SGML文書構造に基づいており、同一のSGML文書構造を有するSGMLテキストに対して共通に利用する。「報告書」での「報告内容」の形式すなわち文書構造を固定とするように、SGML文書構造は、SGMLテキストを保持するカラムによって固定とする。これに従って、抽出パラメタ1002は、処理対象のSGMLテキストを保持しているカラムによって固定とする。そこで、テーブル名とカラム名から抽出パラメタ1002を取得できるように、ディクショナリサーバ1107の機能を用いて抽出パラメタ1002を一元管理する。

【0010】抽出処理は次の手順で行なう。

(1)ディクショナリサーバ1107にアクセスし、操作対象のSGMLテキストを保持しているテーブル名とカラム名をもとに抽出パラメタを取得する。

(2)上記の(1)で取得した抽出パラメタを利用して抽出処理を行なう。

上記の手順(1)(2)の実行制御は、組込みモジュール1125が行なう。

【0011】次に、検索処理について説明する。検索要求「発行日が1996年10月15日より最近の報告書のアブストラクト部分を抽出せよ」は、次のように記述する。

```
SELECT extract(contents, 'abstract')
FROM reports
WHERE published_date >: '1996-10-15'
```

なお、上記の例は、SQL3によって記述されている。

【0012】上記検索要求に見合うデータベース処理は、次のような手順で行なう。

(1)カラムpublished_dateに設定されたインデクスを

用いて、WHERE句の条件を満たすレコードの集合をテーブルreportsから取得する。

(2)上記の(1)で得られたレコードの集合を基にSGMLテキストを逐次取り出し、ADT関数extract()を実現する外部関数p_sgml_extract()をコールして、アブストラクト部分を抽出する。

【0013】上記の(2)でSGMLテキストを逐次取り出してアブストラクト部分を抽出する処理については、並列処理機能を有効に利用して高速に処理すること及び要求受付サーバ1105に検索結果をまとめるときのデータ転送量を少なくすることを考慮して、実行サーバ1106で行なう。すなわち、実行サーバ1106では、アブストラクト部分を抽出するために外部関数p_sgml_extract()をコールし、組込みモジュール1125に実行制御を渡す。実行制御が渡された組込みモジュール1125は、ディクショナリサーバ1107にアクセスして抽出パラメタを取得する。そして、その抽出パラメタを利用して抽出処理を行なう。

【0014】なお、データベース管理システム(DBMS)に関する他の従来技術として次のものがある。

(1)並列データベース処理

例えば、特開平8-137910号公報に記載されているように、ユーザの問い合わせを処理プロセッサが受け付け、複数のデータベースエンジンプロセッサで最適に負荷が分散されるようにDBMSが実行制御を行なう。

(2)SQL3

SQL3は、国際標準化機構ISOが標準化をすすめているデータベース言語仕様の草案であり、例えば「Information technology - Database languages - SQL - Part 2: SQL/Foundation"ISO/IEC JTC1/SC21 N10489」に記載されている。

【0015】(3)オブジェクト・リレーショナルDBMS

オブジェクト指向の概念を取り入れたリレーショナルDBMSである。例えば「オブジェクトリレーショナルDBMSs, Michael Stonebraker著、大田佳伸訳、インターナショナル・トムソン・パブリッシング・ジャパン1996年8月」に記載されており、SQL3のADTおよびADT関数に対応付けることができるユーザ定義型およびユーザ定義関数を使用可能にすべきであることが述べられている。次の例は、ユーザ定義の複合型phone_tの定義文である。

```
create type phone_t (
    area        varchar(3),
    number      varchar(7),
    description  varchar(20));
```

上記の例は、複合型phone_tが、areaという名前で参照される3バイト以下の可変文字列型の要素と、numberという名前で参照される7バイト以下の可変文字列型の要素と、descriptionという名前で参照される20バ

イト以下の可変文字列型の要素の3つの要素から構成されることを示している。次の例は、ユーザ定義関数Northness_equal

```
create function Northness_equal (point, point) returns Boolean
with selfunc = selectivity_comp
external name '/usr/Northness_equal'
language C;
```

上の例は、ユーザ定義関数Northness_equal() が、C言語で記述された外部関数 /usr/Northness_equal に関連付けられていることを示している。

【0016】(4)関数定義の指定

例えば「共通オブジェクトリクエスト・ブローカ ～構造と仕様～、オブジェクトマネジメントグループ編著、相磯秀夫監訳、社団法人 日本オフィスオートメーション協会、原本“The Common Object Request Broker:Architecture andSpecification” OMG Document Number 9.1.12.1, Revision 1.1」に記述されているようなIDL (Interface Definition Language:インタフェイス定義言語)を用いる関数定義の指定方法がある。この方法では、CORBAと呼ばれるソフトウェアアーキテクチャにおいて、モジュール間のインタフェイスをIDLによって定義する。そのインタフェイスをC言語などのプログラミング言語に対応付け、スタブと呼ばれる接続用のモジュールを生成する。このスタブモジュールを介することによって、柔軟なモジュール間通信を可能としている。

【0017】

【発明が解決しようとする課題】前記従来のデータベースシステム1000では、次のような問題がある。すなわち、前記検索要求に見合うデータベース処理では、並列処理する複数の実行サーバ1106上の組込みモジュール125の全てから1つのディクショナリサーバ1107に対するアクセスが発生し、ディクショナリサーバ1107で抽出パラメタを取り出す処理を集中して行なうことになる。このため、ディクショナリサーバ1107の負荷が一時的に過大となり、ディクショナリサーバ1107の処理性能がネックとなって、システム性能が低下してしまう。そこで、本発明の第1の目的は、上記の問題を解消し、システム性能を向上させることが出来るデータベース管理方法および並列データベース管理システムを提供することにある。

【0018】また、前記IDLの仕様では、組込みモジュールの実行制御を行なうことを直接指定することができない問題がある。そこで、本発明の第2の目的は、組込みモジュールの実行制御を行なうことを直接指定可能するデータベース管理方法を提供することにある。

【0019】

【課題を解決するための手段】第1の観点では、本発明は、ユーザからの要求を受け付けてその要求に見合うデータベース処理の実行手順を指示する実行手順コードを作成する要求受付サーバと前記実行手順コードに従って

hness_equal() の定義文である。

前記データベース処理を並列に実行する複数の実行サーバとを有し、ユーザによって組み込まれる組込みモジュールを実行する機能を前記要求受付サーバと前記実行サーバとが有する並列データベース管理システムにおいて、前記実行サーバ上で組込みモジュールを実行する際に必要な入力情報を前記要求受付サーバが取得すべきことを認識する解析過程と、前記認識に従って前記要求受付サーバ上で入力情報取得用の組込みモジュールを実行して前記入力情報を取得する取得過程と、前記入力情報を渡す指示を含ませるように前記実行手順コードを編集する設定過程とを含むデータベース実行制御を行なうことを特徴とするデータベース管理方法を提供する。上記第1の観点によるデータベース管理方法では、複数の実行サーバ上の組込みモジュールが必要とする入力情報のうち、いずれの実行サーバでも同一のものは、それぞれの実行サーバが取得するのではなく、要求受付サーバが取得し、それをそれぞれの実行サーバが利用する。このため、入力情報を取得する処理の重複や集中がなくなり、システム性能の低下を防止できる。

【0020】第2の観点では、本発明は、上記構成のデータベース管理方法において、組込みモジュールのインタフェイス仕様の記述をもとに組込みモジュールの実行制御を行なうことを特徴とするデータベース管理方法を提供する。上記第2の観点によるデータベース管理方法では、組込みモジュールの実行制御をインタフェイス仕様の記述をもとに行うが、このインタフェイス仕様の記述をユーザが行えるから、結局、組込みモジュールの実行制御を行なうことをユーザが直接指定できることとなる。そして、インタフェイス仕様の記述に、ユーザによる組込みモジュールの機能呼び出しで渡される入力情報以外の付加的な入力情報を当該組込みモジュールの実行のために渡す指示または実行サーバ上で組込みモジュールを実行する際に必要な入力情報を要求受付サーバが取得すべきことの指示の少なくとも一方を含ませれば、前記第1の観点によるデータベース管理方法を好適に実施できる。

【0021】第3の観点では、本発明は、上記構成のデータベース管理方法において、ユーザによる組込みモジュールの機能呼び出しで渡される入力情報以外の付加的な入力情報を当該組込みモジュールの実行のために渡す指示を当該組込みモジュールのインタフェイス仕様の記述に含み得ることを特徴とするデータベース管理方法を提供する。上記第3の観点によるデータベース管理方法では、組込みモジュールの実行制御のもととなるインタ

フェイス仕様の記述に、ユーザによる組込みモジュールの機能呼び出しで渡される入力情報以外の付加的な入力情報を当該組込みモジュールの実行のために渡す指示を含ませるから、前記第1の観点によるデータベース管理方法を好適に実施できる。

【0022】第4の観点では、本発明は、上記構成のデータベース管理方法において、前記実行サーバ上で組込みモジュールを実行する際に必要な入力情報を前記要求受付サーバが入力情報取得用の組込みモジュールを実行して取得すべきことを示す指示を前記実行サーバ上で実行する組込みモジュールまたは前記入力情報取得用の組込みモジュールの少なくとも一方のインタフェース仕様の記述に含み得ることを特徴とするデータベース管理方法を提供する。

【0023】上記第4の観点によるデータベース管理方法では、組込みモジュールの実行制御のもととなるインタフェース仕様の記述に、実行サーバ上で組込みモジュールを実行する際に必要な入力情報を要求受付サーバが取得すべきことの指示を含ませるから、前記第1の観点によるデータベース管理方法を好適に実施できる。

【0024】第5の観点では、本発明は、ユーザからの要求を受け付ける要求受付サーバとその要求受付サーバの指示に従ってユーザからの要求に見合うデータベース処理を並列に実行する複数の実行サーバとを有する並列データベース管理システムにおいて、前記要求受付サーバが入力情報取得用の組込みモジュールを実行して入力情報を取得し、前記実行サーバが前記入力情報を用いて組込みモジュールを実行することを特徴とする並列データベース管理システムを提供する。上記第5の観点によるデータベース管理システムでは、複数の実行サーバ上の組込みモジュールが必要とする入力情報のうちで、いずれの実行サーバでも同一のものは、それぞれの実行サーバが取得するのではなく、要求受付サーバが取得し、それをそれぞれの実行サーバが利用する。このため、入力情報を取得する処理の重複や集中がなくなり、システム性能の低下を防止できる。

【0025】

【発明の実施の形態】以下、図を用いて本発明の実施の一形態を詳細に説明する。図1は、本発明にかかるデータベースシステムの構成図である。このデータベースシステム100は、データベースを利用するアプリケーションプログラム101と、データベース管理システム102と、記憶装置126、128および140とを具備している。

【0026】前記アプリケーションプログラム101は、一般的なデータベースシステムの運用で行われているように、以下の手順にて作成される。

(1)ユーザがデータベース言語（この例ではSQLとす）によって問合せ文記述する。

(2)上記の(1)の記述を入力とし、データベース管理シ

ステム102が提供するアプリケーションプログラム作成ツールを用いて、問合せ処理を要求するプログラムのソースコードを生成する。

(3)上記の(2)のソースコードを適当なプログラミング言語コンパイラでコンパイルし、アプリケーションプログラム101を得る。

【0027】前記アプリケーションプログラム101は、データベース管理システム102が提供する問合せ処理要求関数103をコールする。この問合せ処理要求関数103をコールするときに、ユーザ問合せ104を入力とする。そして、問合せ要求関数103は、データベース管理システム102（の要求受付サーバ105の問合せ処理制御部109）に、ユーザの問合せを伝える。

【0028】前記データベース管理システム102は、前記アプリケーションプログラム101からの要求を受け付ける要求受付サーバ105と、ユーザからの要求に見合うデータベース処理を並列に実行する複数の実行サーバ106と、システムの定義情報を管理するディクショナリサーバ107と、これらのサーバ105、106、107を接続する通信路108と、組込みモジュール定義登録ツール132とを具備している。

【0029】前記要求受付サーバ105は、前記アプリケーションプログラム101からの要求を受け付けて当該要求に見合う処理の実行制御を行なう問合せ処理制御部109と、受け付けた問合せ処理要求を解析する問合せ解析部110と、問合せ解析部110の解析結果をもとにしてユーザの要求に見合ったデータベース処理の実行手順を指示する実行手順コード111および実行サーバ106で処理を実行するときに入力情報として渡す実行時入力情報112を実行用ワークエリア113に作成する実行手順コード作成部114と、前記実行手順コード111および前記実行時入力情報112をもとに実行サーバ106にデータベース処理を行なうように指示する実行指示部115と、入力情報取得用の組込みモジュール117と、前記実行指示部115の指示に応じて実行サーバ106が返却する処理結果を保持する実行結果保持部118とを具備している。

【0030】前記実行指示部115は、組込みモジュール117を実行して前記実行時入力情報112を取得する処理の制御を行なうパラメタバインド制御部116を含んでいる。なお、パラメタバインド制御部116によって組込みモジュール117を実行して前記実行時入力情報112を決定することを「パラメタバインド」と呼ぶ。前記組込みモジュール117は、このパラメタバインドを行なうコードを保持している。

【0031】前記組込みモジュール定義登録ツール132は、組込みモジュール117、125のインタフェース仕様を記述するための組込みモジュールインタフェース定義言語を提供し、この組込みモジュールインタフェ

イス定義言語によって記述された組込みモジュール仕様記述131を解析し、その解析結果をもとに組込みモジュール定義情報130をディクショナリサーバ107に登録する。

【0032】前記実行サーバ106は、前記要求受付サーバ105の実行指示部115からの指示を受信しその指示に従った処理を行なうように制御する実行制御部119と、前記要求受付サーバ105の実行指示部115から送信される実行手順コード111および実行時入力情報112を実行手順コード123および実行時入力情報124として保持するための実行用ワークエリア120と、前記記憶装置126に格納しているデータベースファイル121への入出力および実行サーバ106上でのデータの取り扱いを管理するためのデータアクセス管理部122と、実行サーバ106において実行する外部関数からなる組込みモジュール125とを具備している。

【0033】前記ディクショナリサーバ107は、データベースシステム100に関する定義情報を管理する定義情報管理部127を具備している。前記定義情報は、記憶装置128に格納されており、ユーザ定義関数情報129と組込みモジュール定義情報130とからなる。これらユーザ定義関数情報129および組込みモジュール定義情報130の詳細については、後で図4を参照して説明する。

【0034】このデータベースシステム100におけるユーザの問合せは、次の手順で処理される。

- (1)アプリケーションプログラム101からの要求を要求受付サーバ105が受け付ける。
- (2)要求受付サーバ105が要求に見合った処理を実行サーバ106に指示する。
- (3)実行サーバ106が要求に見合った処理を実行し、実行結果を要求受付サーバ105の実行結果保持部118に通信路108を介して直接送信する。これにより処理結果が要求受付サーバ105の実行結果保持部118に保持される。

【0035】(4)アプリケーションプログラム101からの処理結果取り出しの要求を要求受付サーバ105が受け付ける。

- (5)要求受付サーバ105が、実行結果保持部118で保持している実行結果をアプリケーションプログラム101に返す。

【0036】図2は、問合せ処理制御部109がアプリケーションプログラム101からの問合せ要求に応じて行なう問合せ処理の流れ図である。この問合せ処理の入力はユーザ問合せ104であり、出力および処理結果は実行結果保持部118に実行結果を保持させることである。ステップ201では、問合せ処理制御部109は、問合せ解析部110を用いてユーザ問合せ104を解析する。この解析の結果、実行手順コードパターン選定結

果を取得する。このステップ201の問合せ解析処理の詳細については、後で図3を参照して説明する。ステップ202では、問合せ処理制御部109は、前記ステップ201で取得した実行手順コードパターン選定結果を基に実行手順コード作成部114を用いて実行手順コード111を作成する。この実行手順コード111の詳細については、後で図5を参照して説明する。また、このステップ202の実行コード作成処理の詳細については、後で図6を参照して説明する。ステップ203では、問合せ処理制御部109は、前記ステップ202で取得した実行手順コード111を基に実行指示部115を用いて実行サーバ106にユーザの要求に見合った処理を行なうように指示する。そして、処理を終了する。このステップ203の実行指示処理の詳細については、後で図7を参照して説明する。

【0037】図3は、前記ステップ201の問合せ解析処理を示す流れ図である。この問合せ解析処理(201)の入力はユーザ問合せ104であり、出力および処理結果は実行手順コードパターン選定結果とパラメタバインド指示情報リストである。ステップ301では、問合せ解析部110は、ユーザ問合せ104の構文解析および意味解析を行なう。すなわち、ユーザ問合せ104の中でADT関数がコールされているか、どんな型のデータを操作対象としているか等が解析される。このステップ301の処理は、一般的なSQLの構文解析および意味解析の手法をもとにSQL3の記述に対応するよう拡張した形態でよい。ステップ302では、問合せ解析部110は、前記ステップ301によって解析した結果をもとにディクショナリサーバ107の定義情報管理部127にアクセスし、ユーザ定義関数情報129および組込みモジュール定義情報130を参照し、ADT関数と外部関数の対応を調べて、組込みモジュールの実行形態を解析する。このステップ302が、「実行サーバ上で組込みモジュールを実行する際に必要な入力情報を要求受付サーバが取得すべきことを認識する解析過程」に相当する。ステップ303では、問合せ解析部110は、前記ステップ302の解析結果をもとにユーザの要求に見合ったデータベース処理の実行手順コードパターンを選定し、実行手順コードパターン選定結果を生成する。そして、処理を終了する。

【0038】図4は、ユーザ定義関数情報129および組込みモジュール定義情報130の構成図である。ユーザ定義関数情報129は、ADT関数と外部関数との対応付けを保持するADT関数-外部関数テーブル401と、ADT関数パラメタの構成情報を保持するADT関数パラメタテーブル402とからなる。前記ADT関数-外部関数テーブル401は、ADT関数名、ADT型名および外部関数名を示すカラムからなる。ADT関数-外部関数テーブル401のレコード403は、ADT関数extract()が、sgmltext_t型に定義され、外部関数

p_sgml_extract()に対応付けられることを表わしている。このレコード403を参照することにより、問合せ解析部110は、ユーザ問合せ104においてADT関数extract()を実行するよう要求されている場合、外部関数p_sgml_extract()をコールしなければならない、という解釈を行なう。前記ADT関数パラメータテーブル402は、ADT関数名、パラメータ番号、パラメータ型を示すカラムからなる。ADT関数パラメータテーブル402のレコード404は、ADT関数extract()の第1パラメータのパラメータ型がsgmltext_tであることを表わしている。ここで、ADT関数の第1パラメータで受け渡すパラメータ値は、当該ADT関数が処理対象とするADT型のデータとする。ADT関数パラメータテーブル402のレコード405は、ADT関数extract()の第2パラメータのパラメータ型がVARCHARであることを表わす。意味的には、ADT関数extract()の第2パラメータで受け渡すパラメータ値は、抽出部分を指定するタグの名称である。すなわち、ADT関数extract()は、この第2パラメータで指定されたタグによって区切られたSGML文書の部分を抽出する。前記ADT関数パラメータテーブル402のレコード404およびレコード405を参照することにより、問合せ解析部110は、ユーザ問合せ104におけるADT関数extract()のsgmltext_t型およびVARCHAR型の2つのパラメータを解析する。

【0039】組込みモジュール定義情報130は、組込みモジュール117、125に含まれる外部関数とその外部関数に関する修飾情報を保持する外部関数テーブル406と、組込みモジュール117、125に含まれる外部関数のパラメータに関する情報を保持する外部関数パラメータテーブル407とからなる。前記外部関数テーブル406は、外部関数名および関数修飾を示すカラムからなる。外部関数テーブル406のレコード408は、関数修飾のカラム値に外部関数修飾子「ADT関数対応」を保持することにより、外部関数p_sgml_extract()がADT関数に対応付けられた外部関数であることを示している。外部関数テーブル406のレコード409は、関数修飾のカラム値に外部関数修飾子「パラメータバインド」を保持することにより、外部関数p_sgml_load_sgmldef()がパラメータバインドを行なうための外部関数であることを示している。

【0040】前記外部関数パラメータテーブル407は、外部関数名、パラメータ番号、パラメータ型およびパラメータ修飾を示すカラムからなる。外部関数パラメータテーブル407のレコード410は、外部関数p_sgml_extract()の第1パラメータがパラメータ型sgmltext_tのパラメータ値を受け渡すことを示している。外部関数パラメータテーブル407のレコード411は、外部関数p_sgml_extract()の第2パラメータがパラメータ型VARCHARのパラメータ値を受け渡すことを示している。なお、問合せ解析部110は、上記の外部関数p_sgml_extract()の第1パラメータお

よび第2パラメータについては、それぞれADT関数の第1パラメータおよび第2パラメータのパラメータ値をそのまま受け渡す、という解釈を行なう。これは、前記ADT関数-外部関数テーブル401のレコード403、前記ADT関数パラメータテーブル402のレコード404およびレコード405を参照することにより、外部関数p_sgml_extract()がADT関数extract()と対応付けられているからである。

【0041】外部関数パラメータテーブル407のレコード412は、外部関数p_sgml_extract()の第3パラメータがパラメータ型BLOBのパラメータ値を受け渡すことを示している。また、パラメータ修飾のカラム値にパラメータ修飾子「返却」を保持することにより、当該第3パラメータのパラメータ値を当該関数の返却値として受け渡すことを示している。外部関数パラメータテーブル407のレコード413は、外部関数p_sgml_extract()の第4パラメータがパラメータ型BLOBのパラメータ値を受け渡すことを示している。また、パラメータ修飾のカラム値にパラメータ修飾子「パラメータバインド p_sgml_load_sgmldef」を保持することにより、外部関数p_sgml_load_sgmldef()によってパラメータバインドを行なう、すなわち、外部関数p_sgml_load_sgmldef()が返却する値を当該パラメータ値とすることを示している。外部関数パラメータテーブル407のコード414は、外部関数p_sgml_load_sgmldef()の第1パラメータがパラメータ型BLOBのパラメータ値を受け渡すことを示している。また、パラメータ修飾のカラム値にパラメータ修飾子「返却」を保持することにより、当該第1パラメータのパラメータ値を当該関数の返却値として受け渡すことを示している。外部関数パラメータテーブル407のレコード415は、外部関数p_sgml_load_sgmldef()の第2パラメータがパラメータ型DBCOLUMNINFOのパラメータ値を受け渡すことを示している。ここで、パラメータ型DBCOLUMNINFOは、データベース管理システム102が提供する、カラムに関する定義情報を表すためのデータ型である。パラメータ型DBCOLUMNINFOが指定された場合は、問合せ処理制御部109が、そのパラメータ値として、処理対象となっているカラムの定義情報を渡すように処理する。

【0042】機能的には、外部関数p_sgml_extract()は、抽出パラメータを入力として、指定されたタグの部分を抽出する関数である。また、外部関数p_sgml_load_sgmldef()は、パラメータ型DBCOLUMNINFOのカラム情報からテーブル名、カラム名を取得して、ディクショナリサーバ107にアクセスして抽出パラメータを得る関数である。このデータベースシステム100のユーザは、sgmltext_t型をカラム型として指定するときに、カラムに関する情報と対応させて、抽出パラメータをディクショナリサーバ107に登録する。

【0043】問合せ解析部110は、前記レコード403、レコード408およびレコード410を参照するこ

とにより、外部関数p_sgml_extract()を実行サーバ106で実行させる、という解釈を行なう。これは、ADT関数でADT型データの操作を行なう場合は、データを格納している実行サーバ106で当該ADT関数を実行することが並列処理機能を有効に利用できる、という考えに基づくものである。また、問合せ解析部110は、前記レコード409を参照することにより、外部関数p_sgml_load_sgmldef()を要求受付サーバ105で実行させる、という解釈を行なう。これは、外部関数p_sgml_load_sgmldef()がパラメタバインドを行なうための外部関数であるからである。

【0044】図5は、実行手順コード111の構成図である。この実行手順コード111は、次のSQL3による問合せ文に対応する処理の手順である。

```
SELECT extract(contents, 'abstract')
FROM   reports
WHERE  published_date > '1996-10-15'
```

すなわち、実行手順コード111は、インデックス検索して得られた一時的な結果の集合の要素について外部関数p_sgml_extract()を逐次実行するという、ユーザ問合せ104に見合った処理の実行手順を示している。

【0045】実行手順コード111は、ノード木構造で構成される。このようなノード木構造データは、後述する実行手順コード木構造データ作成処理601(図6)において作成される。ノード501は、当該ノードを説明する情報として「当該実行手順コードは、実行サーバで実行するコードである」ことを示す値を保持している。また、ノード501は、実行内容を示すノードへのポインタを保持している。そのポインタは、ノード502を指している。ノード502は、当該ノードを説明する情報として「左下位ノードを1回だけ実行し、その後は右下位ノードのみを実行する」ことを示す値を保持している。ノード502の左下位ノードはノード503であり、右下位ノードはノード506である。ノード503は、当該ノードを説明する情報として「一時的な処理結果リストを生成する」ことを示す値を保持している。ノード503の左下位ノードはノード504であり、右下位ノードはノード505である。ノード504は、ノード503における一時的な処理結果リストを識別するためのリストIDを保持する。ノード505は、当該ノードを説明する情報として「インデックスを利用した検索を行なう」ことを示す値を保持している。結局、上記ノード503、504および505は、「インデックスを利用した検索を行い、検索結果を結果リストIDで識別される一時結果リストとする」ことを示している。ここでは、テーブルreportsのカラムpublished_dateに設定されたインデックスを用いて、published_date > '1996-10-15'の条件を満たすレコードの検索を行なうことが指示される。そして、検索の結果、条件を満たすレコードのレコードIDの集合が一時結果リストとして得られる。

【0046】ノード506は、当該ノードを説明する情報として「左下位ノードの処理結果を右下位ノードに渡す」ことを示す値を保持している。ノード506の左下位ノードはノード507であり、右下位ノードはノード508である。ノード507は、当該ノードを説明する情報として「(当該実行手順コードに従う処理での一時結果リストから) リスト要素を一件取り出す」ことを示す値を保持している。ここでは、リスト要素はレコードIDである。ノード508は、当該ノードを説明する情報として「(ノード506から渡されるリスト要素に対応する) レコード値をデータベースファイルから取り出す」ことを示す値を保持している。ここでは、レコードIDに対応するレコード値を取り出す。

【0047】前記ノード508の下位ノードであるノード509は、当該ノードを説明する情報として「(ノード508がデータベースファイルから取り出したレコード値の一部である) カラム値を取り出す」ことを示す値を保持している。ここでは、テーブルreportsのレコード値のカラムcontentsに保持しているカラム値、すなわちSGMLテキストを取り出す。前記ノード509の下位ノードであるノード510は、当該ノードを説明する情報として「(ノード509で取り出したカラム値に対して) 外部関数を実行する」ことを示す値を保持している。ここでは、カラムcontentsに保持しているSGMLテキストに対して外部関数p_sgml_extract()を実行する。なお、ノード510は、外部関数p_sgml_extract()の実行時の入力として実行用ワークエリア113に保持する実行時入力情報112に保持する入力パラメタ値511へのポインタを保持している。ここでは、要求受付サーバ105のパラメタバインドによって設定した抽出パラメタを入力パラメタ値511としてポイントする。パラメタバインドとポインタの設定については、後で図8を参照して詳述する。

【0048】図6は、前記ステップ202の実行手順コード作成処理の流れ図である。この実行手順コード作成処理(202)の入力は問合せ解析部110が出力する実行手順コードパターン選定結果であり、出力および処理結果は実行手順コード111とパラメタバインド指示情報リストである。ステップ601では、実行手順コード作成部114は、実行手順コードパターン選定結果からノード木構造データを作成する。この実行手順コード木構造データ作成処理は、データベース処理の実行手順を示す一般的な制御情報と同等のものが作成できればよい。なお、ステップ601で作成されるノードには、外部関数コールを指示するノードも含まれる。ステップ602では、実行手順コード作成部114は、パラメタバインドの実行を指示するパラメタバインド指示情報リストを作成する。すなわち、外部関数コールを指示するノードを探し、外部関数コールを指示するノードを見つけたら、そのパラメタを探して、パラメタバインドが

指定されていないか確認する。パラメタバインドが指定されていれば、その定義情報をもとにパラメタバインドを実行するための指示情報を作成する。例えば、図5のノード木構造データを探索して、外部関数p_sgm1_extract()を呼び出すためのノード510を見つける。次に、図4の外部関数パラメタテーブル407から外部関数p_sgm1_extract()をコールするときの第4パラメタの入力として外部関数p_sgm1_load_sgmldef()を実行してパラメタバインドを行なわなければならない、という解釈を行なう。これらの情報をもとに、「ノード510の外部関数実行の第4パラメタ入力に外部関数p_sgm1_load_sgmldef()を実行して得られる返却値を設定せよ」というパラメタバインド指示情報を作成できる。このようなパラメタバインド指示情報が実行手順コードに対して複数存在する場合は、リスト形式で管理し、パラメタバインド指示情報リストを作成する。

【0049】図7は、前記ステップ203の実行指示処理の流れ図である。この実行指示処理(203)の入力は前記実行手順コード作成部114が出力する実行手順コード111とパラメタバインド指示情報リストであり、出力および処理結果は実行結果保持部118に実行結果を保持させることである。ステップ701では、実行指示部115は、実行手順コード111とパラメタバインド指示情報リストをもとに、パラメタバインド制御部116を用いてパラメタバインドを実行する。パラメタバインド制御部116を用いたパラメタバインドの実行の詳細については、後で図8を参照して説明する。ステップ702では、実行指示部115は、パラメタバインドを行なった実行手順コード111をもとに、ユーザの要求に見合う処理が複数の実行サーバ106で並行に実行されるように、それぞれの実行サーバ106に処理実行を指示する。また、実行サーバ106へ処理を指示する時に、要求受付サーバ105の実行用ワークエリア113の情報をそのまま実行サーバ105の実行用ワークエリア120に送信する。

【0050】図8は、前記ステップ701のパラメタバインド処理の流れ図である。このパラメタバインド処理(701)の入力は実行手順コード作成部114で作成されたパラメタバインド指示情報リストであり、出力および処理結果はパラメタバインドを実行し入力パラメタへのポインタを実行手順コード111に設定することである。ステップ801では、パラメタバインド制御部116は、パラメタバインド指示情報リストの集合からパラメタバインド指示情報を1つ取り出す。ステップ802では、パラメタバインド制御部116は、取り出したパラメタバインド指示情報をもとにパラメタ値を返却する外部関数を実行する。ここでは、外部関数p_sgm1_load_sgmldef()を実行し、ディクショナリサーバ107から抽出パラメタを取得する。このステップ802が、「要求受付サーバ上で入力情報取得用の組込みモジュール

を実行して入力情報を取得する取得過程」に相当する。ステップ803では、パラメタバインド制御部116は、返却されたパラメタ値を実行用ワークエリア113に実行時入力情報112としてコピーし、パラメタバインド指示情報の指示に従って、この実行時入力情報112をパラメタ値としてポイントするように実行手順コード111に情報を設定する。このステップ803が、「入力情報を渡す指示を含ませるように実行手順コードを編集する設定過程」に相当する。ステップ804では、パラメタバインド制御部116は、パラメタバインド指示情報リストに次のパラメタバインド指示情報が存在するかチェックする。次のパラメタバインド指示情報が存在する場合は前記ステップ801に戻り、次のパラメタバインド指示情報が存在しない場合は処理を終了する。

【0051】以上の処理とデータ構造とにより、要求受付サーバ105で組込みモジュールに含まれる外部関数をコールしてパラメタバインドを行ない、そのパラメタ値を入力として実行サーバ106で組込みモジュールに含まれる外部関数をコールすることが可能となる。

【0052】次に、組込みモジュール定義情報130をディクショナリサーバ107に登録する処理について説明する。図9は、組込みモジュール定義登録ツール132が組込みモジュール定義情報130に登録する処理の流れ図である。この組込みモジュール定義登録処理の入力は組込みモジュール仕様記述131であり、出力および処理結果はディクショナリサーバ107に組込みモジュール定義情報130を保持させることである。ステップ901では、組込みモジュール定義登録ツール132は、組込みモジュール仕様記述131を解析する。データベース管理システム102が提供する組込みモジュールインタフェース定義言語によって記述された組込みモジュール仕様記述131の例を次に示す。この例は、図4に示した組込みモジュール定義情報130に対応するモジュール仕様記述である。

```
p_sgm1_extract (
  in      sgmltext_t,
  in      VARCHAR,
  returns BLOB,
  in      BLOB setter('p_sgm1_load_sgmldef')
) as ADT_FUNCTION;
p_sgm1_load_sgmldef (
  returns BLOB,
  in      DECOLUMNINFO
) as BIND_PARAMETER;
```

上記の記述例では、外部関数p_sgm1_extract()と外部関数p_sgm1_load_sgmldef()とをデータベース管理システム102から呼び出すためのインタフェース仕様が示されている。

【0053】実行サーバ106で実行される外部関数p_

sgml_extract()のインタフェース仕様において、キーワードreturns は、パラメタの修飾子であり、当該パラメタが返却用に用いられることを示す。また、キーワードsetterに続いて外部関数名を指定することにより、当該パラメタがパラメタバインドの対象であることを示し、ここに指定された外部関数によってパラメタ値を取得することを示す。また、キーワードasに続いて関数修飾子ADT_FUNCTION を指定することにより、当該関数がADT関数に対応付けられた外部関数であることを示す。キーワードreturns から as ADT_FUNCTION: までの記述が、「ユーザによる組込みモジュールの機能呼び出しで渡される入力情報以外の付加的な入力情報を当該組込みモジュールの実行のために渡す指示」に相当する。また、キーワードsetterに続いて外部関数名を指定することが、「実行サーバ上で組込みモジュールを実行する際に必要な入力情報を要求受付サーバが入力情報取得用の組込みモジュールを実行して取得すべきことを示す指示」に相当する。なお、ユーザからの要求に用いられるADT関数extract() のインタフェース仕様では、ADT型sgmltext_tとパラメタ型VARCHAR の2つの入力をとるように定義されているのに対して、この2つの入力以外にパラメタ型BLOBの入力を2つとるように外部関数p_sgml_extract()が定義されている。

【0054】次に、要求受付サーバ105で実行される外部関数p_sgml_load_sgmldef() のインタフェース仕様において、キーワードreturns は、パラメタの修飾子であり、当該パラメタが返却用に用いられることを示す。また、キーワードasに続いて関数修飾子BIND_PARAMETER を指定することにより、当該関数がパラメタバインドに用いられる外部関数であることを示す。キーワードreturns から DBCOLUMNINFO までの記述が、「ユーザによる組込みモジュールの機能呼び出しで渡される入力情報以外の付加的な入力情報を当該組込みモジュールの実行のために渡す指示」に相当する。また、キーワードasに続いて関数修飾子BIND_PARAMETERを指定することが、「実行サーバ上で組込みモジュールを実行する際に必要な入力情報を要求受付サーバが入力情報取得用の組込みモジュールを実行して取得すべきことを示す指示」に相当する。

【0055】ステップ902では、組込みモジュール定義登録ツール132は、解析結果を組込みモジュール定義情報130としてディクショナリサーバ107の定義情報管理部127を用いて登録する。そして、処理を終了する。

【0056】

【発明の効果】本発明のデータベース管理方法および並列データベース管理システムによれば、複数の実行サーバ上の組込みモジュールが必要とする入力情報のうちで、いずれの実行サーバでも同一のものは、それぞれの実行サーバが取得するのではなく、要求受付サーバが取得し、それをそれぞれの実行サーバが利用するので、入力情報を取得する処理の重複や集中を防止でき、システム性能を向上できる。また、組込みモジュールの実行制御を、組込みモジュールのインタフェース仕様の記述において指示できるようになる。

【図面の簡単な説明】

【図1】本発明の一実施形態にかかるデータベースシステムの構成図である。

【図2】図1のデータベースシステムにおける問合せ処理の流れ図である。

【図3】図2の問合せ処理における問合せ解析処理の流れ図である。

【図4】図1のデータベースシステムにおけるユーザ定義関数情報と組込みモジュール定義情報の構成図である。

【図5】図1のデータベースシステムにおける実行手順コードの構成図である。

【図6】図2の問合せ処理における実行手順コード作成処理の流れ図である。

【図7】図2の問合せ処理における実行指示処理の流れ図である。

【図8】図7の実行指示処理におけるパラメタバインド処理の流れ図である。

【図9】図1のデータベースシステムにおける組込みモジュール定義登録処理の流れ図である。

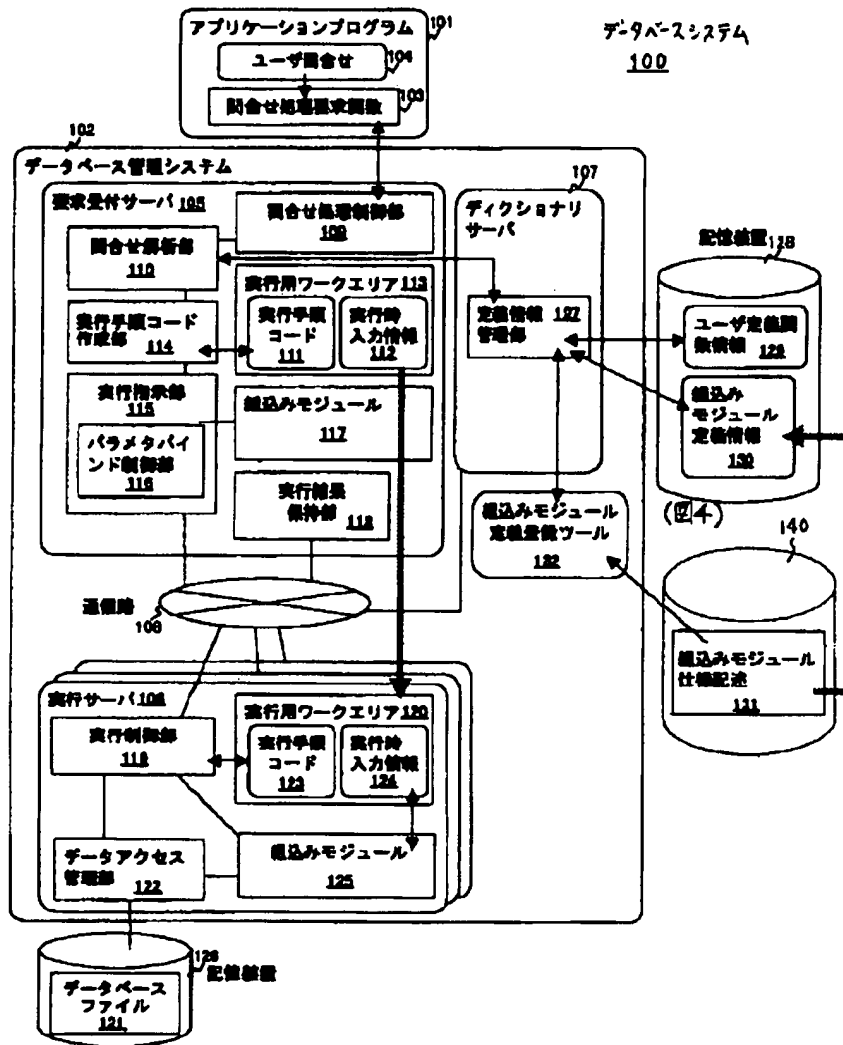
【図10】従来のデータベースシステムの一例の構成図である。

【符号の説明】

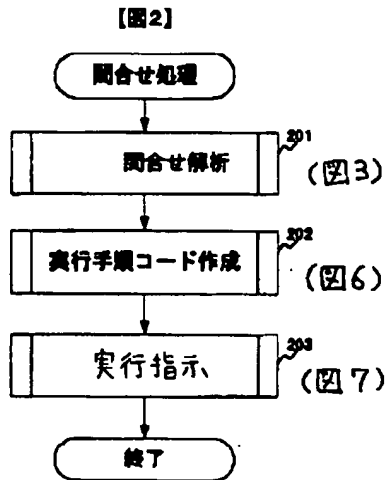
- 100：データベースシステム
- 102：データベース管理システム
- 105：要求受付サーバ
- 106：実行サーバ
- 117：組込みモジュール
- 125：組込みモジュール
- 111：実行手順コード
- 112：実行時入力情報
- 124：実行時入力情報
- 131：組込みモジュール仕様記述

【図1】

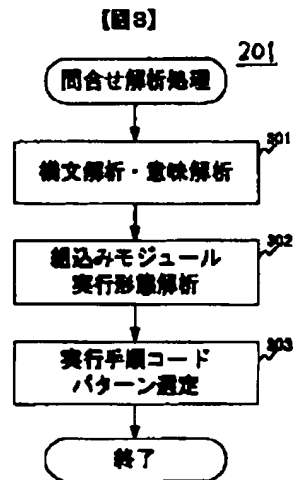
【図1】



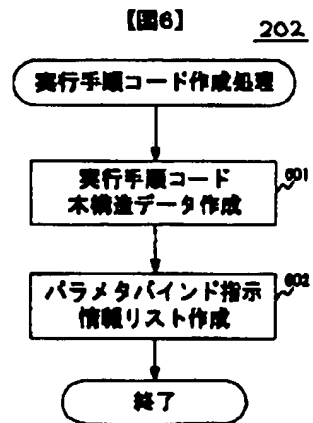
【図2】



【図3】

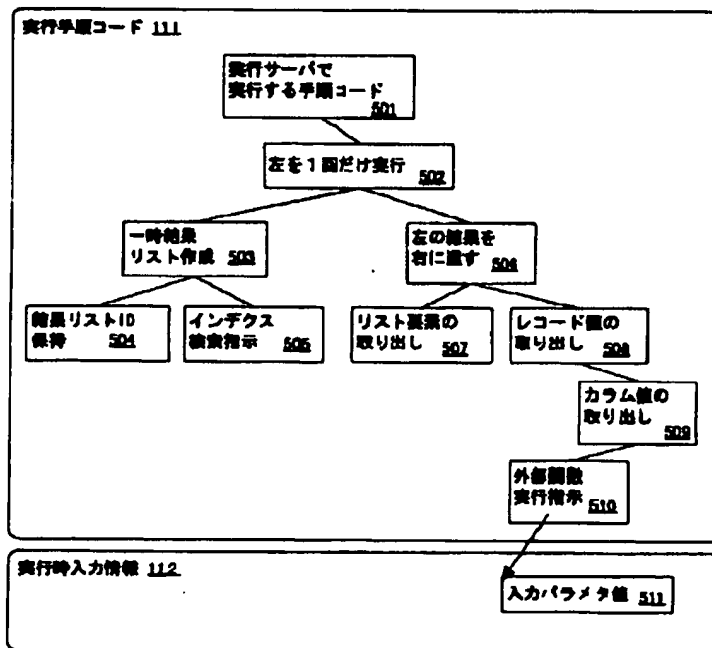


【図6】



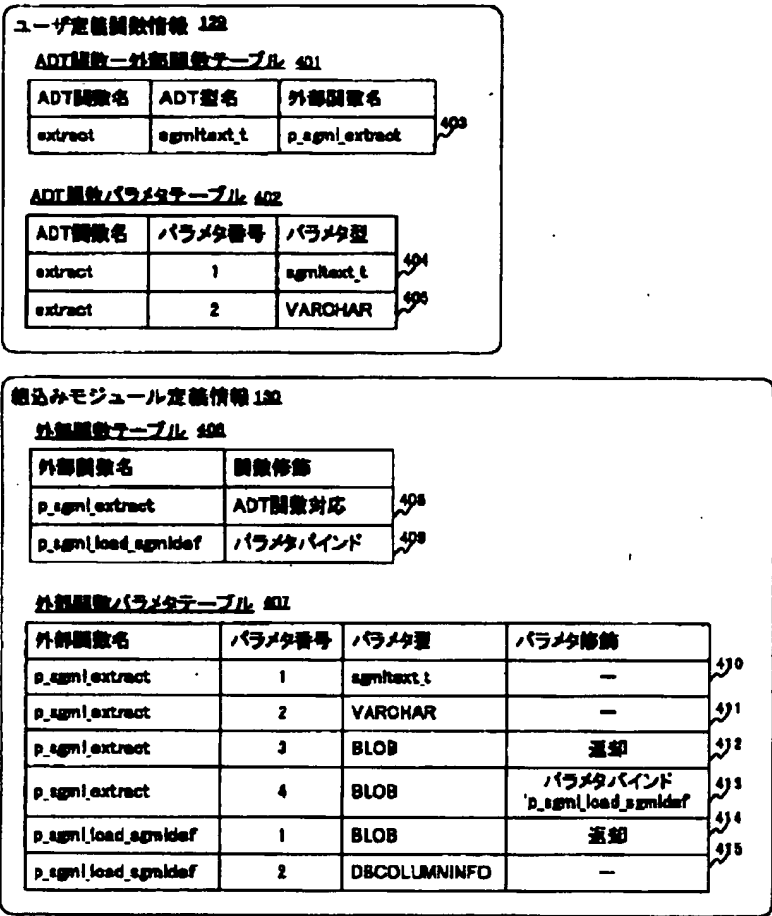
【図5】

【図5】

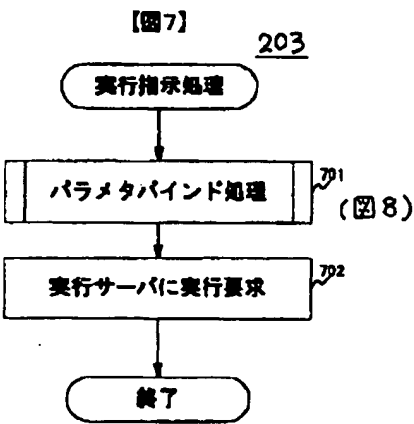


【図4】

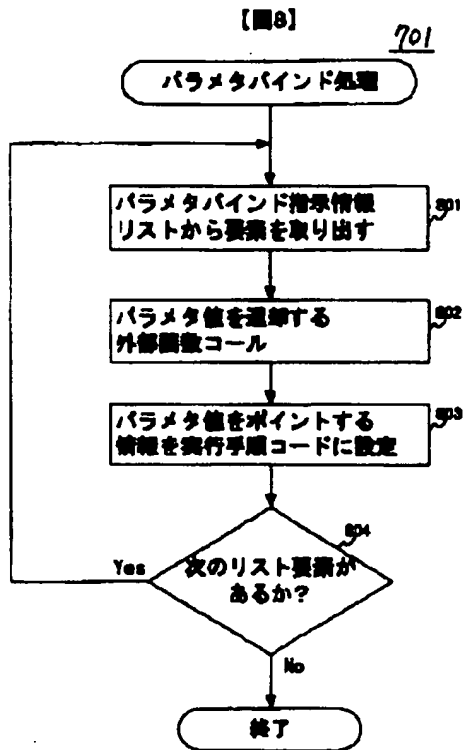
【図4】



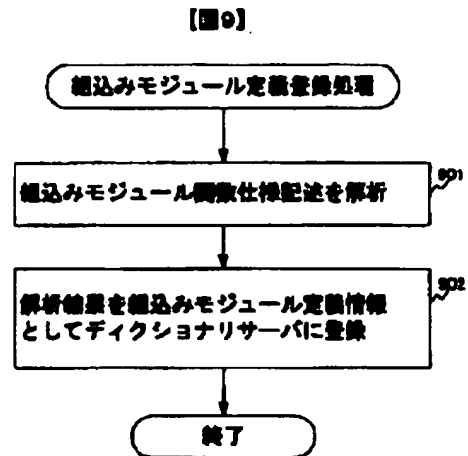
【図7】



【図8】

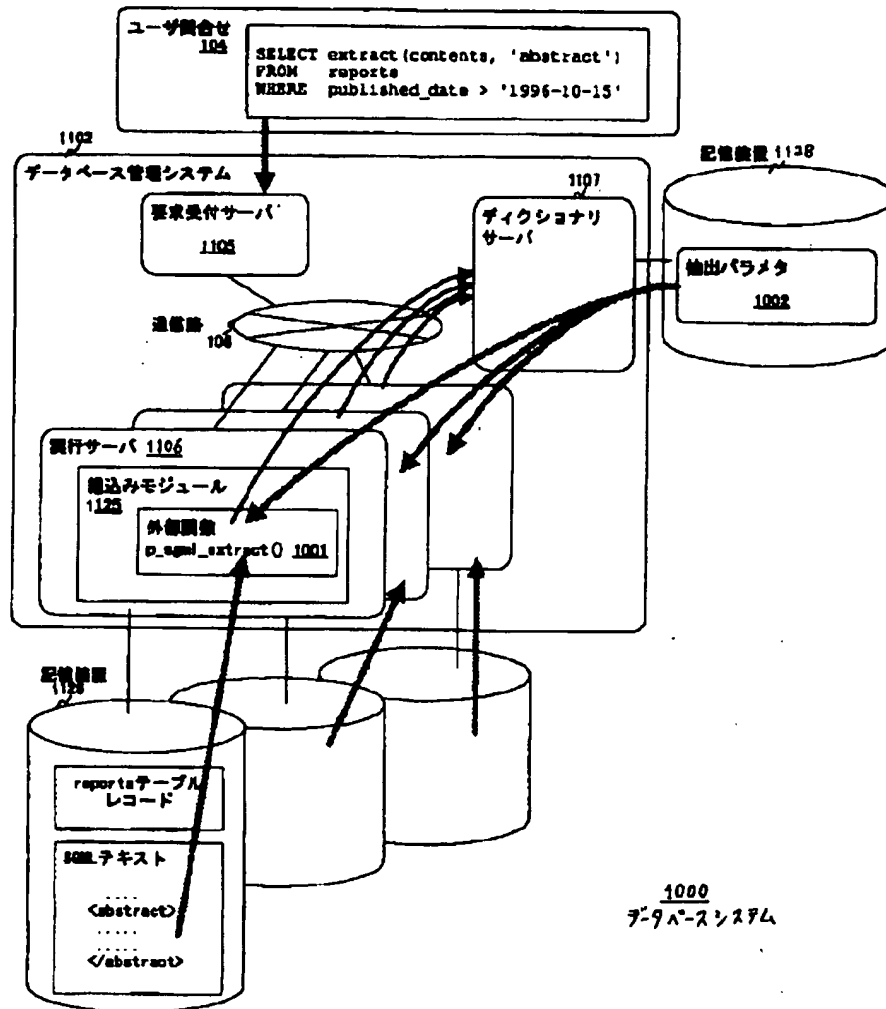


【図9】



【図10】

【図10】



フロントページの続き

(72) 発明者 中野 幸生

神奈川県川崎市幸区鹿島田890番地の12
株式会社日立製作所情報・通信開発本部内

(72) 発明者 原 憲宏

神奈川県川崎市幸区鹿島田890番地の12
株式会社日立製作所情報・通信開発本部内

(72) 発明者 亀城 嘉人

神奈川県横浜市戸塚区戸塚町5030番地 株
式会社日立製作所ソフトウェア開発本部

【公報種別】特許法第17条の2の規定による補正の掲載
 【部門区分】第6部門第3区分
 【発行日】平成14年2月28日(2002. 2. 28)

【公開番号】特開平10-232875
 【公開日】平成10年9月2日(1998. 9. 2)
 【年通号数】公開特許公報10-2329
 【出願番号】特願平9-35021
 【国際特許分類第7版】

G06F 17/30
 12/00 513
 545

【F I】

G06F 15/40 310 B
 12/00 513 D
 545 A
 15/403 330 B

【手続補正書】

【提出日】平成13年8月27日(2001. 8. 27)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】発明の名称

【補正方法】変更

【補正内容】

【発明の名称】 データベース管理方法、並列データベース管理システム、データベース処理方法およびデータベース処理システム

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】 ユーザからの要求を受け付けてその要求に見合うデータベース処理の実行手順を指示する実行手順コードを作成する要求受付サーバと前記実行手順コードに従って前記データベース処理を並列に実行する複数の実行サーバとを有し、ユーザによって組み込まれる組込みモジュールを実行する機能を前記要求受付サーバと前記実行サーバとが有する並列データベース管理システムにおいて、

前記実行サーバ上で組込みモジュールを実行する際に必要な入力情報を前記要求受付サーバが取得すべきことを認識する解析過程と、前記認識に従って前記要求受付サーバ上で入力情報取得用の組込みモジュールを実行して前記入力情報を取得する取得過程と、前記入力情報を渡す指示を含ませるように前記実行手順コードを編集する設定過程とを含むデータベース実行制御を行なうことを

特徴とするデータベース管理方法。

【請求項2】 請求項1に記載のデータベース管理方法において、組込みモジュールのインタフェイス仕様の記述をもとに組込みモジュールの実行制御を行なうことを特徴とするデータベース管理方法。

【請求項3】 請求項2に記載のデータベース管理方法において、ユーザによる組込みモジュールの機能呼び出しで渡される入力情報以外の付加的な入力情報を当該組込みモジュールの実行のために渡す指示を当該組込みモジュールのインタフェイス仕様の記述に含み得ることを特徴とするデータベース管理方法。

【請求項4】 請求項2または請求項3に記載のデータベース管理方法において、前記実行サーバ上で組込みモジュールを実行する際に必要な入力情報を前記要求受付サーバが入力情報取得用の組込みモジュールを実行して取得すべきことを示す指示を前記実行サーバ上で実行する組込みモジュールまたは前記入力情報取得用の組込みモジュールの少なくとも一方のインタフェイス仕様の記述に含み得ることを特徴とするデータベース管理方法。

【請求項5】 ユーザからの要求を受け付ける要求受付サーバとその要求受付サーバの指示に従ってユーザからの要求に見合うデータベース処理を並列に実行する複数の実行サーバとを有する並列データベース管理システムにおいて、前記要求受付サーバが入力情報取得用の組込みモジュールを実行して入力情報を取得し、前記実行サーバが前記入力情報を用いて組込みモジュールを実行することを特徴とする並列データベース管理システム。

【請求項6】 データを処理するデータベース処理方法において、

入力された問い合わせを解析し、

上記解析結果に基づいて、データ処理プログラムへ上記

データを受け渡すかを判定し、
上記判定の結果データを受け渡す場合は、保持した上記
データ参照するためのデータ構造情報を参照し、上記デ
ータ処理プログラムに受け渡すデータを生成し、
上記問合せに基づいたデータ処理指示と上記生成したデ
ータを出力することを特徴とするデータベース処理方
法。

【請求項7】 請求項6に記載のデータベース処理方法
において、
上記出力するデータは、上記データ処理指示と上記生成
したデータとを1つにパッケージして、上記データ処理
プログラムに受け渡すことを特徴とするデータベース処
理方法。

【請求項8】 請求項6に記載のデータベース処理方法
において、
上記データ処理プログラムは、プラグインモジュールを
起動してデータ処理を行うものであり、
上記プラグインモジュールのインターフェースを示すイ
ンターフェース仕様定義に、データ処理プログラムへ上
記生成したデータを受け渡すことを示す指示を含み、
上記指示に基づいて、上記生成したデータをデータ処理
プログラムに受け渡し、
データ処理プログラムが、上記インターフェース仕様定
義に基づいて、プラグインモジュールを起動し、
上記プラグインモジュールが上記生成したデータを参照
してデータ処理を行うことを特徴とするデータベース処
理方法。

【請求項9】 請求項8に記載のデータベース処理方法
において、
上記プラグインモジュールは、データ処理機能を提供す
る関数からなり、
上記関数は、データ入力のパラメタを有し、
上記インターフェース仕様定義は、上記関数とデータ入
力のパラメタの仕様を含み、データ処理プログラムの呼
び出しを含む上記入力した問い合わせを解析し、
その解析結果に基づいて、上記インターフェース仕様定
義を参照して、上記データ処理プログラムの呼び出しに
対応する上記関数を選定し、
上記生成したデータを上記パラメタに設定し、
上記データ処理プログラムは上記選定した関数を呼び出
し、上記生成したデータを関数のパラメタに受け渡すこ
とを特徴とするデータベース処理方法。

【請求項10】 請求項6に記載のデータベース処理方
法において、上記入力した問合せに含まれるデータ処理
プログラムの呼び出しは、文書データの部分抽出を行う
機能の呼び出しであり、
上記データ処理プログラムに受け渡すデータは、文書デ
ータの部分抽出した結果として出力する文書データの構
造情報を示すデータであることを特徴とするデータベ
ース処理方法。

【請求項11】 データを処理するデータベース処理シ
ステムにおいて、入力した問合せを解析する解析手段
と、

上記解析結果に基づいて、データ処理手段へ上記データ
を受け渡すかを判定し、

上記判定の結果データを受け渡す場合は、保持した上記
データ参照するためのデータ構造情報を参照し、データ
処理プログラムに受け渡すデータを生成するデータ生成
手段と、

上記問合せに基づいたデータ処理指示と上記生成した上
記データ処理手段に受け渡すインターフェース手段とを
備えたことを特徴とするデータベース処理システム。

【請求項12】 データを処理するデータベース処理方
法において、

入力された問い合わせを解析し、

その解析結果に基づいて、データ処理プログラムへ上記
データを受け渡すかを判定し、判定結果に応じて上記デ
ータを出力することを特徴とするデータベース処理方
法。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0001

【補正方法】変更

【補正内容】

【0001】

【発明の属する技術分野】本発明は、システム性能を向
 上することが出来るデータベース管理方法、並列データ
 ベース管理システム、データベース処理方法およびデー
 タベース処理システムに関する。

【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0017

【補正方法】変更

【補正内容】

【0017】

【発明が解決しようとする課題】前記従来のデータベ
 スシステム1000では、次のような問題がある。すな
 わち、前記検索要求に見合うデータベース処理では、並
 列処理する複数の実行サーバ1106上の組込みモジュ
 ール125の全てから1つのディクショナリサーバ11
 07に対するアクセスが発生し、ディクショナリサーバ
 1107で抽出パラメタを取り出す処理を集中して行な
 うことになる。このため、ディクショナリサーバ110
 7の負荷が一時的に過大となり、ディクショナリサーバ
 1107の処理性能がネックとなって、システム性能が
 低下してしまう。そこで、本発明の第1の目的は、上記
 の問題を解消し、システム性能を向上させることが出来
 るデータベース管理方法、並列データベース管理シス
 テム、データベース処理方法およびデータベース処理シ
 ステムを提供することにある。

【手続補正5】

【補正対象書類名】明細書

【補正対象項目名】0018

【補正方法】変更

【補正内容】

【0018】また、前記IDLの仕様では、組み込みモジュールの実行制御を行なうことを直接指定することができない問題がある。そこで、本発明の第2の目的は、組み込みモジュールの実行制御を行なうことを直接指定可能するデータベース管理方法、並列データベース管理システム、データベース処理方法およびデータベース処理システムを提供することにある。

【手続補正6】

【補正対象書類名】明細書

【補正対象項目名】0056

【補正方法】変更

【補正内容】

【0056】

【発明の効果】本発明のデータベース管理方法、並列データベース管理システム、データベース処理方法およびデータベース処理システムによれば、複数の実行サーバ上の組み込みモジュールが必要とする入力情報のうち、いずれの実行サーバでも同一のものは、それぞれの実行サーバが取得するのではなく、要求受付サーバが取得し、それをそれぞれの実行サーバが利用するので、入力情報を取得する処理の重複や集中を防止でき、システム性能を向上できる。また、組み込みモジュールの実行制御を、組み込みモジュールのインタフェース仕様の記述において指示できるようになる。